# ARTIFICIAL INTELLIGENCE
# UNIT-I Question-Bank
# Introduction

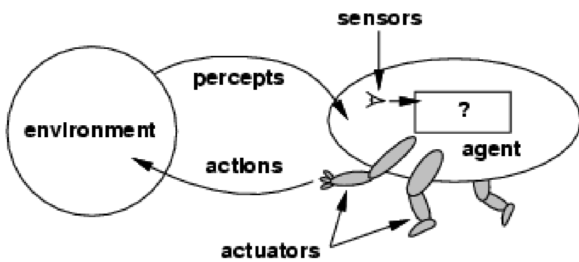
---

**1) What is AI?**

| | |
|---|---|
| Systems that think like humans | Systems that think rationally |
| Systems that act like humans | Systems that act rationally |

**2) Define an agent.**

An **agent** is anything that can be viewed as **perceiving** its **environment** through **sensors** and **acting upon** that environment through **actuators.**




**3) What is an agent function?**

An agent's behavior is described by the **agent function** that maps any given **percept sequence** to an **action.**

**4) Differentiate an agent function and an agent program.**

| Agent Function | Agent Program |
|---|---|
| An abstract mathematical description | A concrete implementation,running on the agent Architecture. |

**5) What can Ai do today?**

- Autonomous Planning and Scheduling
  - Spacecraft control
  - Goal-directed planning, detection, diagnosis, problem recovery
- Game Planing
  - IBM Deep Blue
  - World chess champion
- Autonomous Control
  - CMU NavLab
  - Computer-controlled mini-van
  - Crossed the US without human control over 98% of the time

- Diagnosis
  - Medical diagnosis in several areas of medicine (e.g., pathology)
  - Explanation, justification for decisions
- Logistics Planning
  - DOD's Dynamic Analysis and Replanning Tool
  - Logistics planning of 50,000 vehicles, cargo, people
  - Embarkation, destination, route, conflict resolution
  - Paid back all of DARPA's 30-year investment in AI
- Robotics
  - Robotic surgical assistants
  - Cooperating autonomous robots in reconnaissance
  - Exploration of the Solar System

**6) What is a task environment? How it is specified?**

**Task environments** are essentially the "problems" to which rational agents are the "solutions." A Task environment is specified using PEAS (Performance, Environment, Actuators, Sensors) description.

**7) Give an example of PEAS description for an automated taxi.**

| Agent Type | Performance Measure | Environment | Actuators | Sensors |
|---|---|---|---|---|
| Taxi driver | Safe: fast, legal, comfortable trip, maximize profits | Roads, other traffic, pedestrians, customers | Steering, accelerator, brake, signal, horn, display | Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard |

**Figure 2.4** PEAS description of the task environment for an automated taxi.

**8) Give PEAS description for different agent types.**

| Agent Type | Performance Measure | Environment | Actuators | Sensors |
|---|---|---|---|---|
| Medical diagnosis system | Healthy patient, minimize costs, lawsuits | Patient, hospital, staff | Display questions, tests, diagnoses, treatments, referrals | Keyboard entry of symptoms, findings, patient's answers |
| Satellite image analysis system | Correct image categorization | Downlink from orbiting satellite | Display categorization of scene | Color pixel arrays |
| Part-picking robot | Percentage of parts in correct bins | Conveyor belt with parts; bins | Jointed arm and hand | Camera, joint angle sensors |
| Refinery controller | Maximize purity, yield, safety | Refinery, operators | Valves, pumps, heaters, displays | Temperature, pressure, chemical sensors |
| Interactive English tutor | Maximize student's score on test | Set of students, testing agency | Display exercises, suggestions, corrections | Keyboard entry |

**Figure 2.5** Examples of agent types and their PEAS descriptions.

**9) List the properties of task environments.**

- Fully observable vs. partially observable**.**
- Deterministic vs. stochastic.
- Episodic vs. sequential
- Static vs, dynamic.
- Discrete vs. continuous.
- Single agent vs. multiagent**.**

**10) Write a function for the table driven agent.**

```
function TABLE-DRIVEN-AGENT(percept) returns an action
   static: percepts, a sequence, initially empty
           table, a table of actions, indexed by percept sequences, initially fully specified

   append percept to the end of percepts
   action ← LOOKUP(percepts, table)
   return action
```

**Figure 2.7** The TABLE-DRIVEN-AGENT program is invoked for each new percept and returns an action each time. It keeps track of the percept sequence using its own private data structure.

**11) What are the four different kinds of agent programs?**

Simple reflex agents;
Model-based reflex agents;
Goal-based agents; and
Utility-based agents.

**12) Explain a simple reflex agent with a diagram.**

## Simple reflex agents

The simplest kind of agent is the **simple reflex agent.** These agents select actions on the basis AGENT of the *current* percept, ignoring the rest of the percept history.

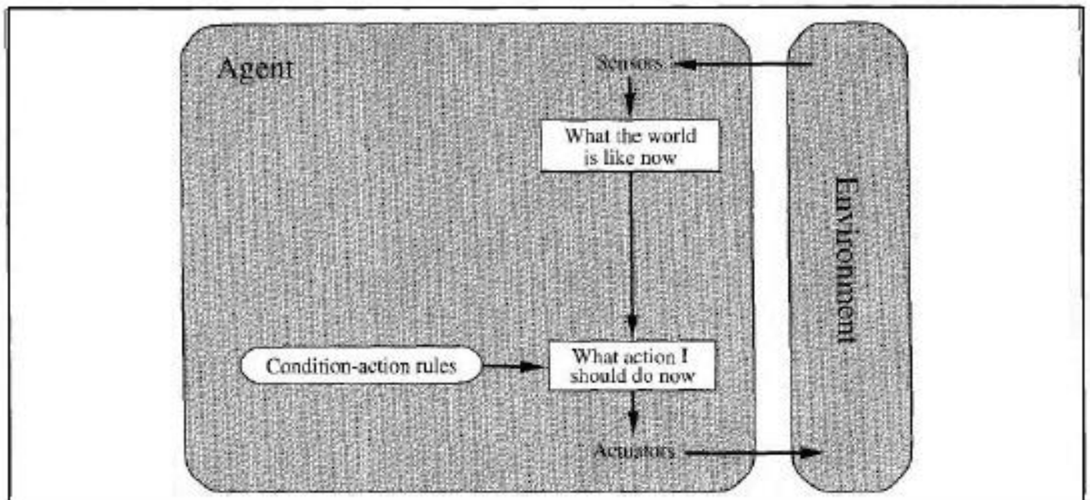



**Figure 2.9** Schematic diagram of a simple reflex agent.

```
function SIMPLE-REFLEX-AGENT(percept) returns an action
   static: rules, a set of condition–action rules

   state ← INTERPRET-INPUT(percept)
   rule ← RULE-MATCH(state, rules)
   action ← RULE-ACTION[rule]
   return action
```

**Figure 2.10** A simple reflex agent. It acts according to a rule whose condition matches the current state, as defined by the percept.

**13) Explain with a diagram the model based reflex agent.**

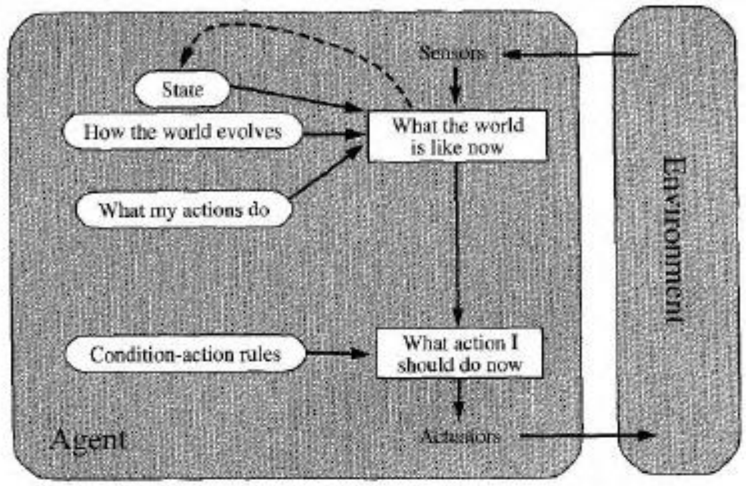



**Figure 2.11** *A* model-based reflex agent.

```
function REFLEX-AGENT-WITH-STATE(percept) returns an action
   static: state, a description of the current world state
           rules, a set of condition–action rules
           action, the most recent action, initially none

   state ← UPDATE-STATE(state, action, percept)
   rule ← RULE-MATCH(state, rules)
   action ← RULE-ACTION[rule]
   return action
```

**Figure 2.12** A model-based reflex agent. It keeps track of the current state of the world using an internal model. It then chooses an action in the same way as the reflex agent.

**13a) Explain with a diagram the goal based reflex agent.**

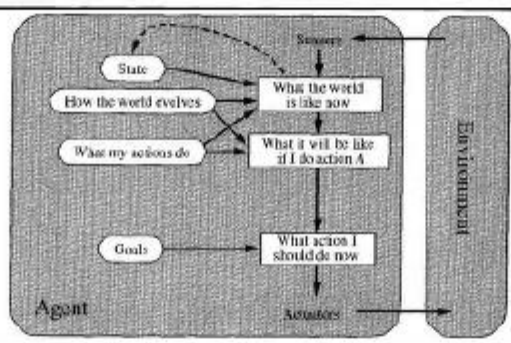



**Figure 2.13** A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.

Knowing about the current state of the environment is not always enough to decide what to do. For example, at a road junction, the taxi can turn left, turn right, or go straight on. The correct decision depends on where the taxi is trying to get to. In other words, as well as a current state description, the agent needs some sort of **goal** information that describes situations that are desirable-for example, being at the passenger's destination.

### 13b) What are utility basedagents?

Goals alone are not really enough to generate high-quality behavior in most environments. For example, there are many action sequences that will get the taxi to its destination (thereby achieving the goal) but some are quicker, safer, more reliable, or cheaper than others.
A **utility function** maps a state (or a sequence of states) onto a real number, which describes the associated degree of happiness.

### 13c) What are learning agents?

A learning agent can be divided into four conceptual components, as shown in Fig- 2.15. The most important distinction is between the learning element, which is re- ELEMENT sponsible for making improvements, and the performance element, which is responsible for selecting external actions. The performance element is what we have previously considered to be the entire agent: it takes in percepts and decides on actions. The learning element uses CRITIC feedback from the critic on how the agent is doing and determines how the performance element should be modified to do better in the future.

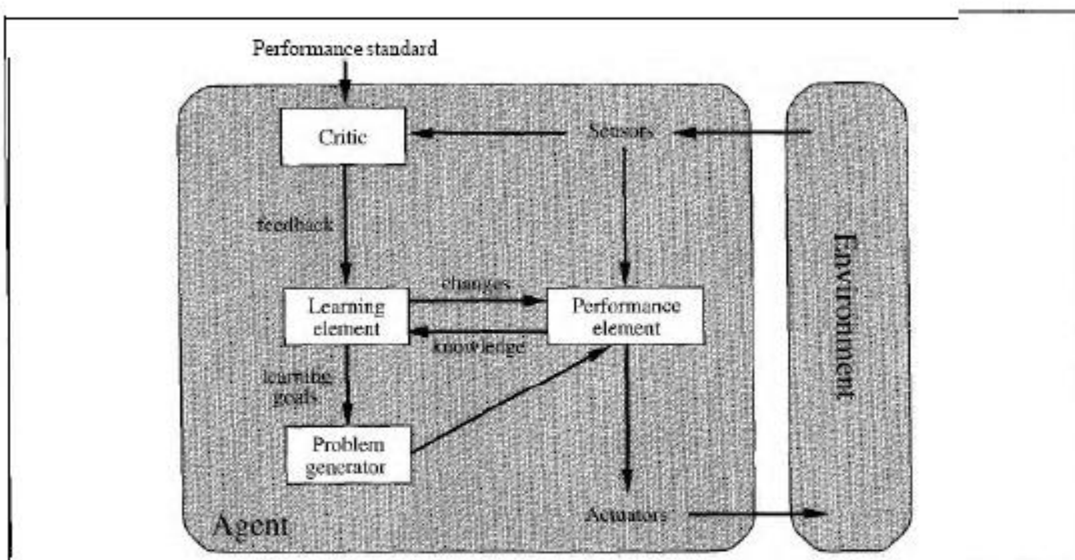



**Figure 2.15** A general model of learning agents.

## Searching Techniques

### 14) Define the problem solving agent.

A Problem solving agent is a **goal-based** agent . It decide what to do by finding sequence of actions that lead to desirable states. The agent can adopt a goal and aim at satisfying it.
Goal formulation is the first step in problem solving.

### 15) Define the terms goal formulation and problem formulation.

**Goal formulation**,based on the current situation and the agent's performance measure,is the first step in problem solving.

The agent's task is to find out which sequence of actions will get to a goal state.

**Problem formulation** is the process of deciding what actions and states to consider given a goal

**16) List the steps involved in simple problem solving agent.**

(i) Goal formulation
(ii) Problem formulation
(iii) Search
(iv) Search Algorithm
(v) Execution phase

**17) Define search and search algorithm.**

The process of looking for sequences actions from the current state to reach the goal state is called **search.**

The **search algorithm** takes a **problem** as **input** and returns a **solution** in the form of **action sequence.** Once a solution is found,the **execution phase** consists of carrying out the recommended action..

**18) What are the components of well-defined problems?**

- The **initial state** that the agent starts in . The initial state for our agent of example problem is described by *In(Arad)*
- A **Successor Function** returns the possible **actions** available to the agent. Given a state x,SUCCESSOR-FN(x) returns a set of {action,successor} ordered pairs where each action is one of the legal actions in state x,and each successor is a state that can be reached from x by applying the action.
  For example,from the state In(Arad),the successor function for the Romania problem would return

{ [Go(Sibiu),In(Sibiu)],[Go(Timisoara),In(Timisoara)],[Go(Zerind),In(Zerind)] }

- Thr **goal test** determines whether the given state is a goal state.
- A **path cost** function assigns numeric cost to each action. For the Romania problem the cost of path might be its length in kilometers.

**19) Differentiate toy problems and real world problems.**

| TOY PROBLEMS | REAL WORLD PROBLEMS |
|---|---|
| A **toy problem** is intended to illustrate various problem solving methods. It can be easily used by different researchers to compare the performance of algorithms. | A **real world problem** is one whose solutions people actually care about. |

**20) Give examples of real world problems.**

(i) Touring problems
(ii) Travelling Salesperson Problem(TSP)
(iii) VLSI layout
(iv) Robot navigation
(v) Automatic assembly sequencing
(vi) Internet searching

**21) List the criteria to measure the performance of different search strategies.**

- **Completeness** : Is the algorithm guaranteed to find a solution when there is one?
- **Optimality** : Does the strategy find the optimal solution?
- **Time complexity** : How long does it take to find a solution?
- **Space complexity** : How much memory is needed to perform the search?

**22) Differentiate Uninformed Search(Blind search) and Informed Search(Heuristic Search) strategies.**

| Uninformed or Blind Search | Informed or Heuristic Search |
|---|---|
| o No additional information beyond that provided in the problem definition<br>o Not effective<br>o No information about number of steps or path cost | o More effective<br>o Uses problem-specific knowledge beyond the definition of the problem itself. |

**23) Define Best-first-search.**

Best-first search is an instance of the general TREE-SEARCH or GRAPH-SEARCH algorithm in which a node is selected for expansion based on the evaluation function f(n ). Traditionally,the node with the *lowest* evaluation function is selected for expansion.